CANONICAL

ubuntu

嵌入式Linux：自制或外购？

2021年6月

两难的选择：选择有商业支持的还是自研？

物联网是一场规模上史无前例的技术变革：到2025年，预计将会有超过300亿的物联网连接1。以5G（第五代移动通信）、LORA（低功耗广域网络）、BT（蓝牙）、NB-IoT（窄带物联网）和ZigBee（紫蜂）等新协议为代表的联网技术和通信技术快速发展，从而推动了这场变革。摩尔定律也是物联网发展的关键因素；规格适合嵌入的计算机硬件越来越强大。物联网高速创新得以成为现实。

新一代应用处理器片上系统（SoCs）正在取代受限的微控制器单元。配备内存和联网功能的芯片可以嵌入设备与系统，部署不受位置限制。如果这类芯片接入互联网，可快速应用于消费者装置、工业自动化、汽车、机器人、物流、数码标牌等诸多领域。

Linux可以说是利用新一代计算机硬件的最佳操作系统。Linux采用宽松免版税许可模式，进而成为了嵌入系统开发的主流操作系统2。社区给予大力支持，不断贡献出适合各种CPU架构的新版优质软件和生成工具，使得平台更加丰富。

开发人员能在Linux内核中找到驱动片上系统及控制显示器、传感器、执行器等外设所需的一切驱动程序。Linux发行版还提供现成库，可以执行新的通信协议。然而这些免费提供的各种功能并非唾手可得，还是面临麻烦与难题。

一方面，由社区维护的buildroot和Yocto等编译系统已经对Linux嵌入进行了抽象化和自动化处理，所有开发人员都可以创建发行版。另一方面，在生产和大规模应用中运行嵌入式Linux的企业往往有社区贡献无法满足的要求。嵌入式Linux供应商需要提供专业知识和企业支持来填补这一空缺。

因此，现今设备制造商处于两难境地：对于嵌入项目的内部平台而言，到底是运行自己的Linux发行版，还是采用具备商业支持的嵌入式Linux发行版？

本指南将帮助您作出抉择。

设备上市的主要难题

全球设备制造商竞相打造能够兑现物联网承诺的物联网设备：数据驱动的价值链优化、流程实时可见性、全新数字客户体验。如果设备制造商选择Linux作为设备基础平台，其通常面临着三个主要难题：

1. 编译主板启动的基本代码
2. 软件开发效率
3. 长期维护

我们现在来详细探讨每个难题，先从主板启动开始。

引导器（bootloader）	初始化主板和加载Linux内核的程序。
Linux内核	内核管理CPU、内存、I/O外设等系统资源。内核还可帮助其他程序通过系统调用访问资源。
根文件系统	含有内核完成初始化后运行的库和程序。

嵌入式Linux基本代码的要素：内核、启动资产和根文件系统

构建适配主板的基本代码

引导目标板启动是所有嵌入项目的第一步。主板启动要求包括，编译启动目标板所需的基本代码，使所有外设可被用户空间应用程序寻址。初始工作围绕着配置启动固件、用必需驱动程序编译合适内核、为根文件系统安装必需的库和配置文件。具备可引导的主板、启用所有传感器和外设是这一阶段成功的标准。

开发板制造商在板级支持包（BSP）中提供了此基本代码。然而，根据具体的目标用例，主板可能需要启用传感器和外设。例如，物流应用需要RFID（射频识别），安全应用需要摄像头。在基本代码中为传感器和外设提供所有正确的内核驱动程序、模块或二进制文件对成功至关重要。但是板级支持包并不一定包含以上内容，这就需要做大量工程工作，将以上内容集成入基本代码。

开发人员还需要调整板级支持包，确保板级支持包可以可靠地用于生产，因为制造商通常会优化板级支持包，以便快速进行原型制作。这相当耗费时间，而主板制造商未必是Linux专家，质量可能参差不齐。而且其并非总能理解对主流Linux源代码的关键修改，而这些修改会使维护难度增加，且会影响稳定性、正常运行时间，以及设备安全性。

高效的软件开发

在启动阶段结束后，主板会在Linux上启动。一切就绪，开发人员可以开始生成最终用户应用程序。所有传感器和外设均可被操作系统寻址，所有可用基本依赖项可以在目标板上运行应用程序（例如适用于配有图形处理单元（GPU）和专用集成电路（ASIC）的硬件加速库）。主板引导完成后，开发人员开始生成实际最终用户应用程序。然而，开发环境和生产环境的差异，以及代码的可移植性，可能会严重影响性能。

在永恒之蓝（WannaCry）恶意软件变体感染计算机系统和制造工具之后的整个周末，台湾积体电路制造股份有限公司（台积电）被迫关停旗下多家工厂。

嵌入式软件开发时的开发环境与生产环境是不同的。开发工具链往往托管在个人计算机上，个人计算机可替更受限的目标板编译应用程序，应用程序实际在目标板上运行。在计算能力更强的主机上对应用程序进行交叉编译能够加速迭代。但是，开发环境与生产环境的差异会增加调试和测试的摩擦。

除此之外，代码的可移植性和可重用性也会影响效率。嵌入式软件往往与运行的硬件紧密相连。而且每款主板的处理器架构、内存占用情况、传感器和外设都不同，所以开发人员在新设备上进行编译时，往往需要从零开始。这增加了开发成本，也推迟了上市时间。

震网（Stuxnet）病毒攻击的是数据采集与监视控制系统（SCADA）以及可编程序逻辑控制器（PLC）。据信，该病毒是导致伊朗核计划重大损坏的罪魁祸首。

长期安全维护

在开发出所有功能并经过测试之后，就进入到生产阶段。物联网设备部署在业务关键型和隐私敏感型环境中。因此，安全漏洞不仅可能对最终用户造成极其不利的影响，还可能导致设备制造商声誉严重受损。更重要的是，设备使用寿命通常长达几十年。因此，设备制造商不得不长期为客户提供安全维护。

Triton是一种恶意软件，能够禁用安全仪表系统，给工厂带来挑战。其被称为“世界上最具有杀伤力的恶意软件”。

而问题在于安全威胁会快速演变。因此，设备制造商为确保设备安全，必需在威胁影响设备之前，减少安全漏洞。这意味着内部专门的团队需要跟踪所安装每个软件组件快速演化的漏洞。团队必须结整合开发、安全和运营技能（DevSecOps）；而无论是内部编译还是外包完成，都代价不菲。

工业物联网安全漏洞

除DevSecOps技能之外，还需要物理基础设施来维护分散在各处的设备的安全性。设备制造商须在云和内容发布网络（CDN）基础设施上投资，以便能够以无线方式提供快速的安全更新。此外，远程安全维护还需要操作系统级的复杂工程来支持无线软件的更新能力，因为由社区维护的Linux发行版一般都缺少这种操作功能。

总的来说，长期安全维护需要投入大量的人力资源、基础设施和工程工作。

嵌入式Linux的两种选择：自制或外购

如果采用Linux，设备制造商就面临着上述难题，其大体上有两种选择。他们可以充分利用Linux的免费开源特性，使用Yocto推出自己的发行版，也可以根据需求定制现有的Linux发行版。他们还可以向开源供应商采购嵌入式Linux。这一切都归结为一个问题：是自制还是外购。

我们现在来详细剖析每种选择，了解其对设备制造商的影响。

推出自己的嵌入式Linux

推出定制嵌入式Linux发行版有两种方法：从零开始构建一个全新的发行版，或者定制一个现有发行版。

定制一个现有Linux发行版

这是一种从上至下的方法，该方法首先从现有Linux发行版开始，然后将其从不需要的程序包中剥离。嵌入目标硬件和用户空间配置所需的其他程序包可按需补充。这种方法的优势是可以重用现有发行版的基本代码。然而，这样构建的定制操作系统可能会偏离构建所基于的发行版，从而出现不兼容问题。选择这种方法的设备制造商可能会发现自己拥有的操作系统过于个性化了，必须长期靠自己来维护。如前所述，这样无论人力资源还是基础设施成本都相当高。

yocto PROJECT

Yocto Project（YP）项目是一个开源合作项目，帮助开发人员创建基于定制Linux的系统，且不受硬件架构影响。项目提供了一套灵活的工具和一个空间；在该空间，世界各地的嵌入开发人员可以分享技术、软件栈、配置和最佳做法，它们可以用于创建嵌入式和物联网设备的定制Linux镜像，也可以用于为需要定制Linux操作系统之处创建定制Linux镜像。

BuildRoot
Making Embedded Linux Easy

BuildrootBuildroot是一个简单、高效、易用的工具，可通过交叉编译生成嵌入式Linux系统。为此，Buildroot能够为您的目标生成交叉编译工具链、根文件系统、Linux内核镜像和启动装载。Buildroot可独立用于这些方案的组合（例如，您可以使用现有交叉编译工具链，可以只利用Buildroot编译自己的根文件系统）。

构建适合嵌入式Linux的系统

利用Yocto或Buildroot等编译系统从零开始构建发行版

设备制造商还可以从零开始构建嵌入式Linux发行版。Yocto和Buildroot等编译系统已大大简化此过程。这些编译系统充分利用可重用和可组合脚本，快速构建完整的嵌入式Linux发行版。脚本由软件供应商和主板制造商组成的大型社区提供和维护。

该方法可提供全新的Linux发行版。因此，选择这一方法的设备制造商必需承担文件编制开销和维护成熟操作系统的开销。而且，如果采用了参考设计，引导过程会变得非常简单，而采用定制或衍生主板，这一定会带来更多麻烦，耽误上市时间。

购买商用嵌入式Linux发行版

设备制造商不用自己构建发行版，而是向供应商购买商用Linux发行版。由嵌入式Linux供应商提供Linux发行版，同时提供数年的支持。供应商承诺在商用的操作系统周期内为支持的主板提供高质量基本代码、及时的安全更新和错误修复。

嵌入式Linux供应商拥有广博的专业知识。他们受益于服务多个客户的学习曲线效应。设备制造商可以选择将嵌入式Linux的开发和维护工作外包给供应商。这一方法虽然需要付费，却可以享受供应商的专门知识和工程服务。从供应商处购买，开发人员可以腾出精力专心实现核心业务价值主张。

Ubuntu Core

Wind River Linux

Suse

商业支持的嵌入式Linux发行版，推荐Ubuntu Core、Wind River和Suse。

两种选择的比较

如上所示，两种选择各有优缺点。对于设备制造商而言，每种选择的吸引力取决于具体的业务环境。决定是运行定制的Linux发行版还是向供应商购买的影响因素分为两大类：成本和经营策略。我们来深入定性分析一下这两个方面。

成本角度

自制或外购决策中的相对成本包括所有可避免的成本。这些成本反映了与开发和支持Linux发行版相关的主要活动。下面分析了这些成本项目的相对价值。

相关成本	说明	自研	外购
主板启动	在目标板上启动Linux必需的基本代码	- *作为板级支持包的构成部分提供*	$$$ *付费由供应商启用主板*
用户空间定制	目标电器的配置和应用程序	$$ *满足应用程序运行时需求的定制可能需要大量工作*	- *这通常属于主板适配的范畴*
许可管理	验证及遵守发行版中包括的软件包许可要求	$$ *基本代码中每个单独库要求的许可证符合验证*	- *这通常属于主板适配的范畴*
文档	用文件记载用户和开发人员的操作系统	$$ *必须为开发人员和最终用户创建文档*	- *Linux供应商提供了操作系统文档*
软件质量管理	通过补丁进行持续的错误跟踪和修复	$$$ *这是建立在操作系统之上产品生命周期的持续过程*	$ *由供应商作为重复性服务提供*
安全维护	跟踪、修复，并及时交付安全补丁	$$$ *固定活动，需要Linux安全专家介入*	$ *供应商通常负责长期安全维护*
无线更新	持续交付软件定义的新产品功能	$$$ *设备代理和内容发布网络需要从零开始构建*	$ *Ubuntu Core等发行版的物联网应用商店包括了OTA功能*
客户技术支持	调试和解决操作系统相关的客户支持票证	$$ *这是建立在操作系统之上产品生命周期的持续过程*	$ *供应商通常帮助解决严重的错误*
机会成本	相对于核心活动，为嵌入式Linux分配资源的成本	$$ *对于核心活动并不与嵌入式系统直接相关的公司而言，机会成本可能相当高*	- *从供应商处购买，就能空出资源分配给核心业务活动。*

总体而言，自制方案通常预付成本更低，但存在隐蔽的后期成本。换而言之，推出定制嵌入式Linux发行版难度小，但时间越久，成本越高。现实确实如此，因为维护定制操作系统的负担相当繁重。中长期都需要大量投入。免费开源软件的诱惑可能掩盖了后期必需的投入。

相比之下，选择购买嵌入式Linux的预付成本更高，因为供应商需要支付的预付成本相当高昂。但是，预付成本涵盖了符合性验证、文档和各种用户空间定制等众多消耗人力的活动。供应商还会另外收取安全和错误修复等持续长期活动的周期性费用。

战略因素

除成本以外，战略因素也会影响自制或外购决策。除了短期底线因素以外，高层业务工作计划可能也会影响决策。我们现在从战略角度考虑一下决策。

软件和支持质量对于目标用例是否重要?

工业物联网领域的很多应用程序都关乎关键任务成败。例如，在与制造、智能城市和公用事业相关的用例中，由于软件缺陷导致的停机可能会引发相当高昂的成本和相当严重的安全后果。因此，此类应用程序必须满足严格的可靠性和可用性要求。

对于这类用例，设备制造商从经验丰富的供应商采购嵌入式Linux才是理智之选。很多供应商已经证明其有能力长期提供优质软件和技术支持。他们经验丰富，具备承担可用性和可靠性责任的能力。对于那些在为任务关键型应用程序提供嵌入式Linux方面经验不足的设备制造商来说，承担此类责任可能会带来商业风险。

上市时间是否重要?

全球技术市场竞争日益激烈。在新兴技术市场，提前或者快速行动往往可以带来竞争优势。设备制造商希望在其他竞争者加入竞赛前，率先用创新智能产品抢占市场。为了赢得招标，嵌入式产品的供应商可能必须保持快节奏的交付进度。

这种情况下，将嵌入式Linux开发外包给供应商可能更具吸引力。经验丰富的供应商交付流程已经过尝试、测试，并且可以预测。他们会依照合同推进大规模的客户项目进度计划。相比之下，选择从零开始构建嵌入式Linux发行版的设备制造商面临一个陡峭的学习曲线，所以在实际完成日期方面存在不确定性。

嵌入式Linux对于核心业务有多重要?

设备制造商可能选择在内部进行操作系统开发而不是外购操作系统。从这个意义而言，选择构建自己的Linux发行版类似于一个纵向一体化战略。对于技术公司来说，构建一个定制的Linux发行版可能是搭建宽广技术平台的基础。但是，对于核心并非技术核心型却希望拓展产品组合的企业而言，选择外包可能更加明智。从供应商处购买嵌入式Linux（而不是为此聘请并维持一个专职团队）的企业业务会更加灵活。

外购的时机?

从上面的评估可以看出，不同选择的成本情况和战略前景各不相同。根据设备制造商的规模、成熟度、资产负债表、核心业务、客户或产品，每种选择都有可能是合理的。接下来我们来了解一下设备制造商应何时选择具体方案。

拥有嵌入式软件工作计划的成熟科技公司：运行自己的软件

根据以上评估，选择推出并维护定制的嵌入式Linux发行版成本最高。但如果内部进行嵌入式Linux开发的预期未来现金流超过了启动调试、开发和长期维护定制Linux发行版必需投入的资金，那么从战略因素角度出发，这样投入可能是合理的。规划的嵌入式产品线越丰富，这样的投入就更显合理。

现金储备充足的大公司可能就是这种情况。如果嵌入式系统开发是核心业务，那么构建定制嵌入式Linux发行版就更有价值。否则，因为机会成本的关系，构建定制发行版的投资内部回报率可能不是最高的。如果公司已经有能够接手这类项目的嵌入式开发团队，这种选择就更有意义了。

产品线扩张或者新投资：获取商业支持的嵌入式Linux发行版

随着物联网在市场上愈显重要，老牌企业采用智能联网产品扩大产品组合。初创企业借助创新智能产品服务打入市场。对初创企业或者老牌企业而言，上市速度至关重要，因为多个对手在同时争抢市场份额。从未进行过嵌入式产品大规模商业化和嵌入式产品支持的公司需要逐渐积累经验。考虑到Linux的复杂性，积累专家专门知识的周期较长，需要大量投入。

对新公司而言，从经验丰富的供应商购买嵌入式Linux的做法更加明智。采购Linux而不是构建内部能力，能压低业务线的固定成本。这样单位经济效益更高，但同时初创企业或者老牌企业风险也更低。
而且，将嵌入式Linux开发外包以后，公司可以增加核心活动的投资，最大限度降低机会成本。

多家Linux供应商提供商业支持的嵌入式Linux发行版。例如，Ubuntu Core是一个嵌入式Linux发行版，它针对大规模任务关键型物联网部署进行了优化，将新的嵌入式产品线推向市场的企业和初创设备制造商都采用Ubuntu Core。下文我们将举例说明Ubuntu Core如何帮助设备制造商应对主板启动、开发人员效率和安全维护等难题。

借助Ubuntu Core将设备推向市场

Ubuntu是最受开发人员欢迎的免费开源Linux发行版。受欢迎程度反映在围绕Ubuntu的软件生态系统中，通过Ubuntu直接可用的软件包超过60,000个。因此，Ubuntu是桌面、云上、边缘和嵌入式系统进行数字创新的首选平台。

Canonical作为发布和支持Ubuntu的供应商，发行频率可以预测。Canonical还可为Ubuntu提供长期的安全维护保证。以此为基础，开发人员可以利用Ubuntu进行创新。

Ubuntu Core介绍

Ubuntu Core是针对物联网原生嵌入式系统优化的锁定版Ubuntu。Ubuntu可为通用计算提供最新最好的开源软件，而Ubuntu Core仅含有您为单一用途设备选择的软件包和二进制文件。

Ubuntu Core构建于snap之上，snap是针对Linux的通用应用程序包。利用snap，嵌入式系统具备了安全性、不变性、模块性和可组合性等优点。软件可通过delta进行无线更新，delta在出现问题时会自动回滚。Canonical为Ubuntu Core提供长期支持，提供内核补丁和错误修复，支持时间长达10年。

主板适配是一种服务

Ubuntu Core兼具定制嵌入式Linux发行版的可组合性和商用Linux发行版的可靠性。Canonical可在snap中提供定制的模块化基本代码，设备制造商可按要求装配。此外，供应商还为大规模任务关键型部署提供工具、基础设施和技术支持。我们通过Ubuntu Core来探索一下嵌入式Linux服务的关键元素。

维护了10年的定制Ubuntu LTS内核

智能设备的可用寿命可能为数年。因此，如果采用Linux构建，在设备的整个可用寿命期间，底层内核必需由发布者进行维护。这一点可能难以做到，因为对于非商用Linux发行版，社区不会提供如此长时间的维护。相反，长期支持是Linux供应商的核心价值主张。所以，商用Linux发行版更适合使用寿命较长的设备。

Canonical为Ubuntu LTS 版本提供长期支持（LTS）内核。对于没有专业前沿工作负载或最新硬件需求的客户和业务合作伙伴，最新的LTS“-generic”内核才是最佳选择，比如Ubuntu 20.04 LTS的默认内核，其将维护到2030年4月。

Ubuntu LTS 内核

Canonical针对目标硬件平台启用了定制Ubuntu内核，采用Ubuntu Core构建智能设备的设备制造商可以从中受益。设备制造商可利用最新的Linux内核，确保集成了所需的全部驱动程序。而且，Canonical承诺对定制的设备制造商Ubuntu LTS*内核提供长达10年的维护期。即Canonical会持续交付安全补丁和错误修复，同时确保不存在倒退。

安全启动和根文件系统定制

除内核外，Canonical还可编译和配置启动装载，作为主板启动的一部分。任务关键型应用程序需要具备安全的启动能力，保证设备上运行软件的完整性。在最新版本中，Ubuntu Core支持安全启动能力。Canonical为设备制造商提供专业知识，启用硬件信任根和安全元素，从而在Ubuntu Core上实现安全启动。

除了启动装载之外，根文件系统也需要由专家来构建和配置。很多应用程序用户空间都需要额外的内核模块和库。Canonical可为运行Ubuntu Core设备构建成生产级根文件系统提供专业帮助。此外，还可以重要或注重隐私的应用程序启用全磁盘加密能力。

嵌入式Linux的DevSecOps效率

整个生命周期的工具链

在内核、启动装载和根文件系统就绪的情况下，开发人员只需为成千上万设备生成应用和生产质量系统镜像。Ubuntu Core配有可用于大规模构建和部署嵌入式软件的端到端工具链。该工具链由Snapcraft、Ubuntu-image和Checkbox组成。

凭借以上工具，开发人员可以构建自动化嵌入式系统开发工作流：从应用开始，再将应用整合到镜像中，然后测试硬件。Snapcraft是一个打包应用的工具，用于嵌入和发布到应用商店。利用Ubuntu-image，开发人员可以装配Ubuntu Core的定制系统镜像。Checkbox允许开发人员对硬件进行自动化测试。

工具	说明
Snapcraft	自动化应用构建 将嵌入的应用打包进安全容器 创建应用之间的接口 创建硬件外设接口 将应用部署到snap商店 将应用部署到私有物联网应用商店
Ubuntu-image	合成来自snap的系统镜像 生成系统镜像
Checkbox	自动化设备测试

该综合工具消除了开发与部署之间的差距（开发运营合作）。最重要的是，利用它，嵌入式系统开发人员可以经常作出大规模改动，同时可以将可预测的冲突减至最少。这通过松散耦合的软件（多亏snap）和自动化支持实现。

适合物联网设备的开发人员生态系统

成本和人才是传统考虑因素，设备制造商现在越来越希望围绕产品构建开发人员生态系统。凭借网络系统，强大的生态系统可为最终用户和开发人员创造更多价值。与设备一起使用的丰富软件，让最终用户受益。另一方面，开发人员可以凭借自己的应用，接触新用户，提高收入。这种情况下，设备制造商为第三方开发人员、系统集成者或增值经销商发布应用提供了平台。每个采用Ubuntu Core构建的设备都由一个物联网应用商店，刺激了平台生态系统的发展。

“借助Ubuntu Core和snap构建博世的ctrlX AUTOMATION应用商店，搭建了一个软件定义的工业制造平台，形成了一个开放生态系统，缩短了生产时间，提高了设备整个生命周期的安全性。使用该平台的工业机器制造商可以打破IT与OT之间的传统壁垒，摆脱专有系统的束缚。从而，他们的开发人员能够使用自选的编程语言创建可扩展的定制解决方案，拓展嵌入式设备和控制器的功能，同时还可在该领域进行长期的安全更新。”

–Hans-Michael Krause，物联网和可编程序逻辑控制器管理总监

如欲了解更多信息，请访问 https://ubuntu.com/internet-of-things/appstore

长期安全维护

应用商店和批量设备操作
本领域的批量设备群操作是物联网解决方案商用时需要解决的重大难题。由于设备的关键作用和部署的地理分布，维护大批设备的成本可能很高：修复问题需要时间，停机成本高。

远程运营能够降低成本。由于该领域具有远程更新软件、修复错误和现场修复设备的能力，设备制造商可以更加及时地解决终端客户的问题，从而减少停机时间，并消除高昂的人工修复干预的需要。

Canonical为设备制造商提供基础设施，可以维护该领域运行Ubuntu Core的设备群。基础设施由生成和存储应用程序的应用商店组成。它还包括全球内容发布网络，用于向全球的设备群发送软件。设备制造商可利用此基础设施维护自己设备上的软件，也可以在其设备群上执行设备管理任务。

了解更多信息

欲详细了解Ubuntu Core 20，请访问我们的网站。

您还可以考虑使用如下资料：

- Ubuntu core产品手册
- Ubuntu Core 20介绍——研讨会
- Over-the-air 更新白皮书

欲联系Canonical了解Ubuntu Core，请点击此处。

1. 2019嵌入市场研究——Aspencore
2. 2020物联网开发人员调查重要发现——Eclipse基金会

©Canonical Limited 2021. Ubuntu, Kubuntu, Canonical 及其相关徽标皆为Canonical Ltd.的注册商标。所有其他商标是其各自所有者的财产。本文档中提及的任何信息如有更改，恕不另行通知，Canonical将不对任何此类更改承担任何责任。

Canonical Limited，一家于英格兰和威尔士注册的公司，公司代码110334C，注册办事处： 12-14 Finch Road, Douglas, Isle of Man, IM99 1TT 增值税登记：GB 003 2322 47

ubuntu
Delivered by Canonical